# Teaching, Learning, and Applying Optimization:

## AMPL's Intuitive Modeling Meets the Python 🐍 Ecosystem

## Part II

Filipe Brandão, Robert Fourer

{filipe,4er@ampl.com}

AMPL Optimization Inc.
www.ampl.com — +1 773-336-AMPL

INFORMS Webinar 2023, Wednesday, December 13, 2023 | 12noon-1pm EDT

# Outline

---

**Part II (live demos):**

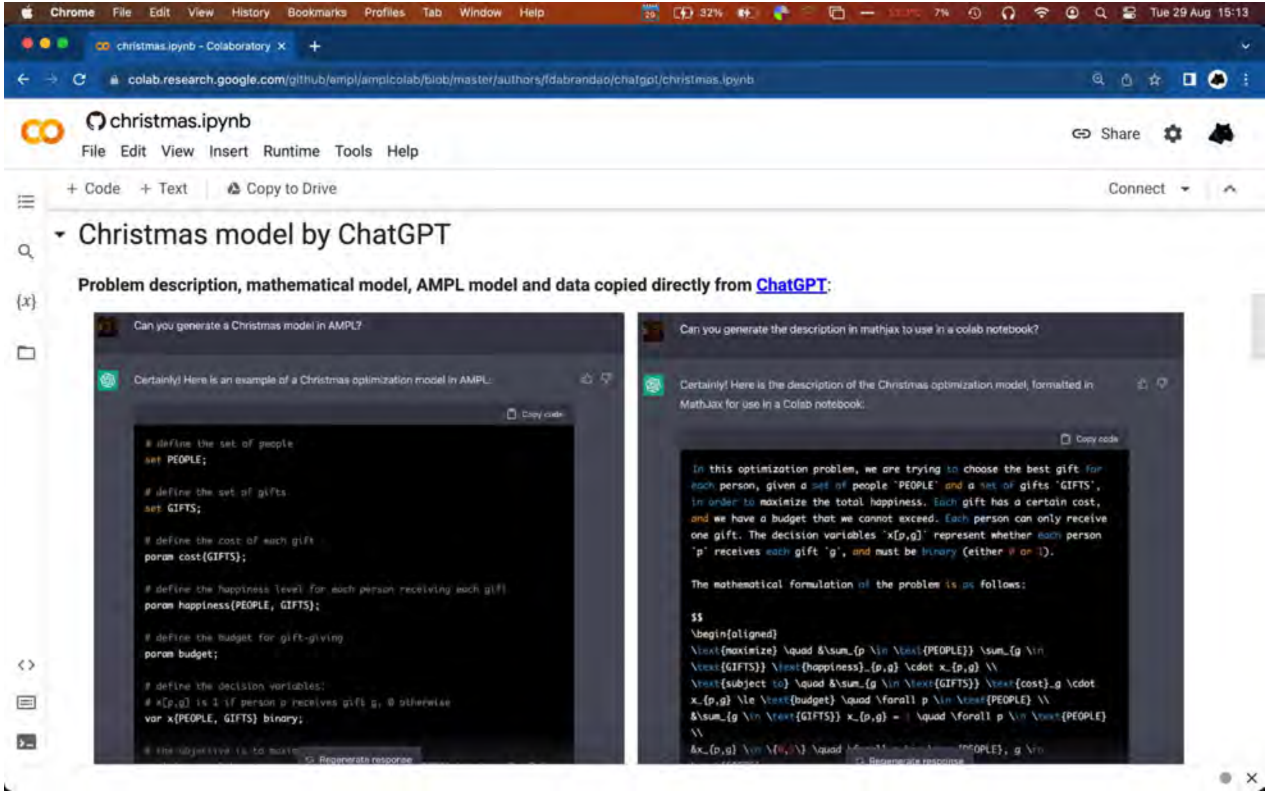
- Quick introduction to amplpy (our Python API)
- AMPL on Google Colab
    - AMPL Model Colaboratory (https://colab.ampl.com)
    - New book: **Hands-On Mathematical Optimization with AMPL in Python** (https://ampl.com/mo-book)
- AMPL and solvers as python packages
- AMPL on Streamlit Cloud
- How to deploy large-scale optimization applications with AMPL

# Quick introduction to amplpy! 🐍

# What do you need to know to use amplpy?

- Basic Python features (lists, dictionaries, etc.)
- Data manipulation with Pandas dataframes
- How to model in AMPL (or how to ask Chat GPT to write AMPL models for you!)

# Example: Christmas model  ([https://colab.ampl.com](https://colab.ampl.com))


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Tue 29 Aug 15:13
christmas.ipynb - Colaboratory
colab.research.google.com/github/ampl/amplcolab/blob/master/authors/fdabrandao/chatgpt/christmas.ipynb
christmas.ipynb
File Edit View Insert Runtime Tools Help
Share
+ Code + Text
Copy to Drive
Connect
Copy code
Copy code
Regenerate response
Regenerate response

## Christmas model by ChatGPT

**Problem description, mathematical model, AMPL model and data copied directly from [ChatGPT](ChatGPT):**

Can you generate a Christmas model in AMPL?

Certainly! Here is an example of a Christmas optimization model in AMPL:

```
# define the set of people
set PEOPLE;

# define the set of gifts
set GIFTS;

# define the cost of each gift
param cost{GIFTS};

# define the happiness level for each person receiving each gift
param happiness{PEOPLE, GIFTS};

# define the budget for gift-giving
param budget;

# define the decision variables:
# x[p,g] is 1 if person p receives gift g, 0 otherwise
var x{PEOPLE, GIFTS} binary;

# the objective is to maxi
```

Can you generate the description in mathjax to use in a colab notebook?

Certainly! Here is the description of the Christmas optimization model, formatted in MathJax for use in a Colab notebook:

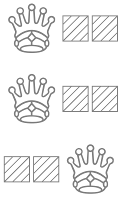
```
In this optimization problem, we are trying to choose the best gift for
each person, given a set of people `PEOPLE` and a set of gifts `GIFTS`,
in order to maximize the total happiness. Each gift has a certain cost,
and we have a budget that we cannot exceed. Each person can only receive
one gift. The decision variables `x[p,g]` represent whether each person
`p` receives each gift `g`, and must be binary (either 0 or 1).

The mathematical formulation of the problem is as follows:

$$
\begin{aligned}
\text{maximize} \quad &\sum_{p \in \text{PEOPLE}} \sum_{g \in
\text{GIFTS}} \text{happiness}_{p,g} \cdot x_{p,g} \\
\text{subject to} \quad &\sum_{g \in \text{GIFTS}} \text{cost}_g \cdot
x_{p,g} \le \text{budget} \quad \forall p \in \text{PEOPLE} \\
&\sum_{g \in \text{GIFTS}} x_{p,g} = 1 \quad \forall p \in \text{PEOPLE}
\\
&x_{p,g} \in \{0,1\} \quad
```

# Example: N-Queens

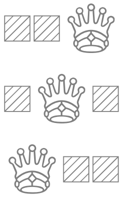
**How can *n* queens be placed on an *n*×*n* chessboard so that no two of them attack each other?**

Constraint **alldiff** enforces a set of integer variables to take distinct values. Using alldiff, we can model N-Queens as follows:

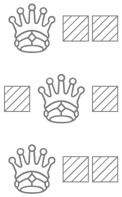
```
param n integer > 0; # N-queens
var Row {1..n} integer >= 1 <= n;
s.t. row_attacks: alldiff ({j in 1..n} Row[j]);
s.t. diag_attacks: alldiff ({j in 1..n} Row[j]+j);
s.t. rdiag_attacks: alldiff ({j in 1..n} Row[j]-j);
```



`Row[1] == Row[2]`
`1 == 1`

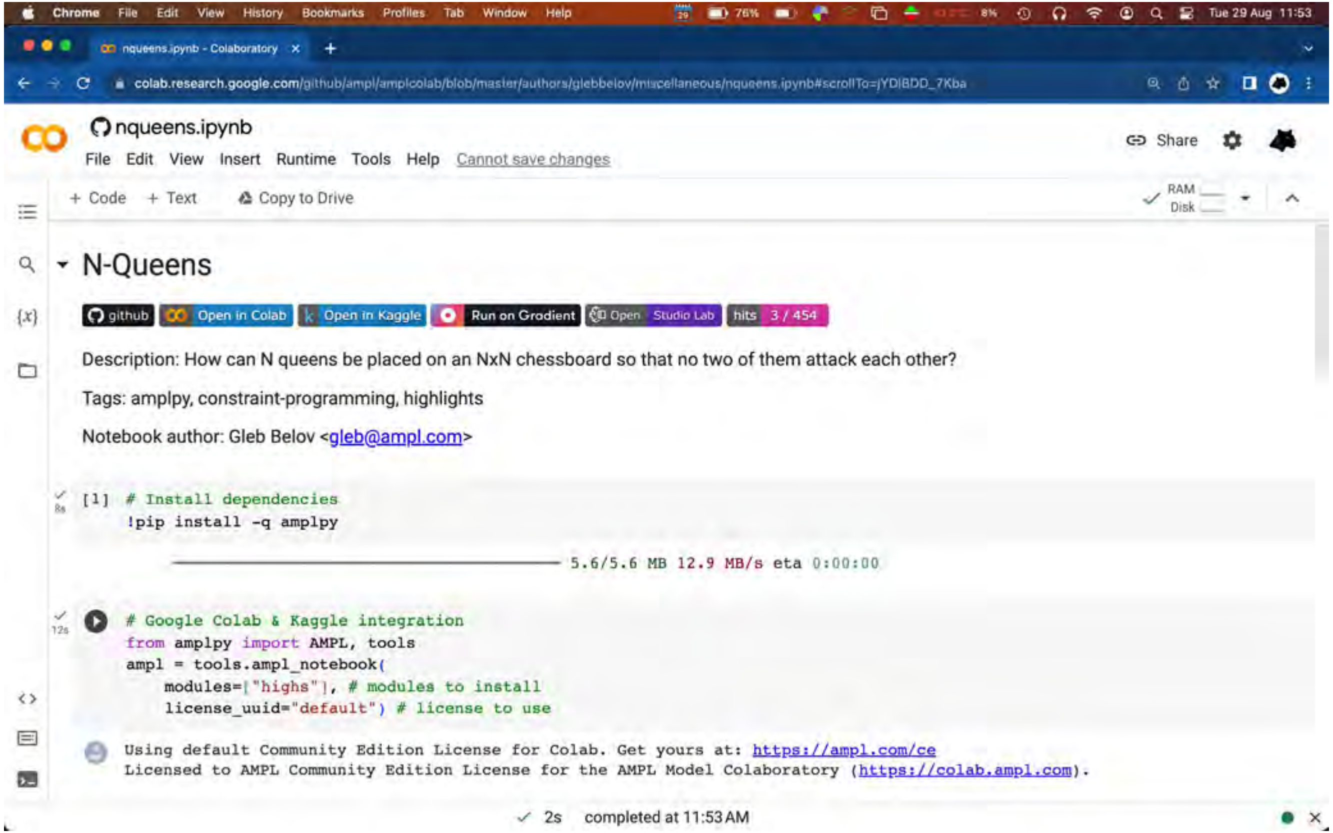


`Row[1]+1 == Row[2]+2`
`3+1 == 2+2`



`Row[1]-1 == Row[2]-2`
`1-1 == 2-2`

# Example: N-Queens ([https://colab.ampl.com](https://colab.ampl.com))


Chrome File Edit View History Bookmarks Profiles Tab Window Help
76%
8%
Tue 29 Aug 11:53
nqueens.ipynb - Colaboratory
colab.research.google.com/github/ampl/amplcolab/blob/master/authors/glebbelov/miscellaneous/nqueens.ipynb#scrollTo=jYDIBDD_7Kba
nqueens.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text Copy to Drive
RAM
Disk
github
Open in Colab
Open in Kaggle
Run on Gradient
Open Studio Lab
hits 3 / 454
8s
12s
✓ 2s completed at 11:53 AM

## N-Queens

Description: How can N queens be placed on an NxN chessboard so that no two of them attack each other?

Tags: amplpy, constraint-programming, highlights

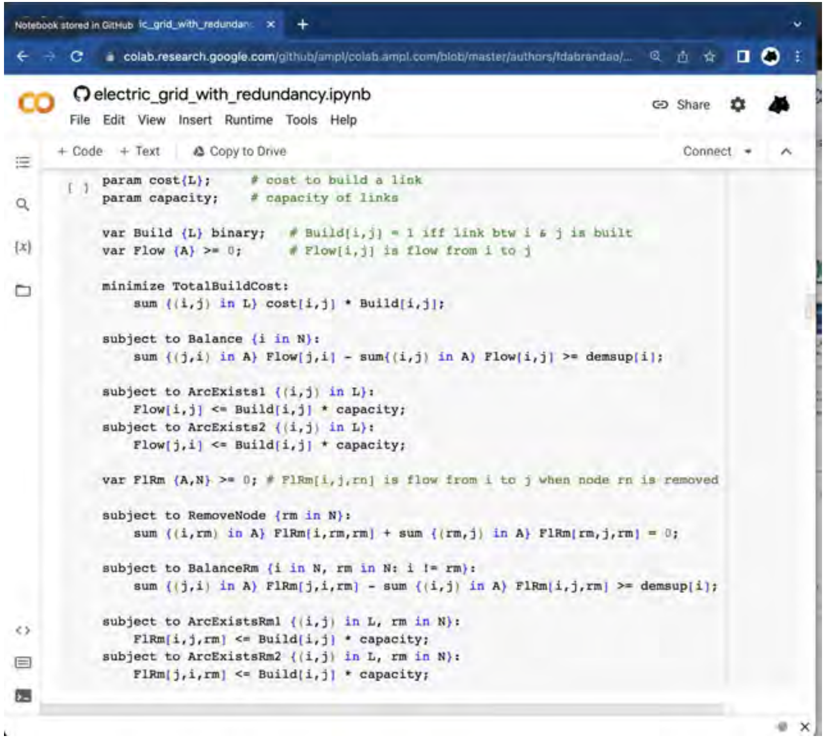
Notebook author: Gleb Belov <gleb@ampl.com>

```
[1] # Install dependencies
    !pip install -q amplpy

        ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.6/5.6 MB 12.9 MB/s eta 0:00:00
```

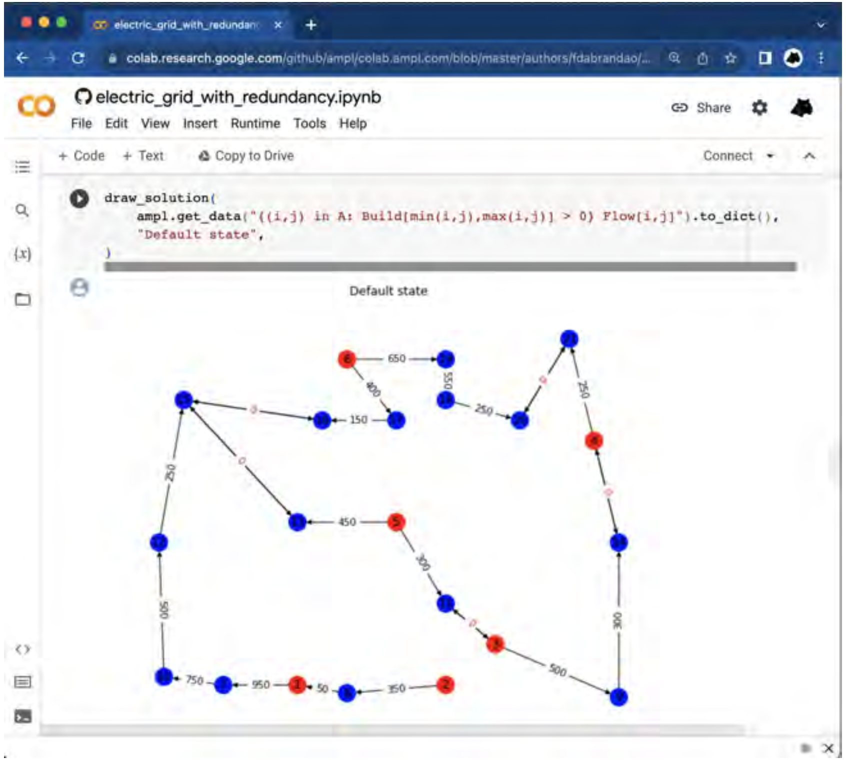
```
# Google Colab & Kaggle integration
from amplpy import AMPL, tools
ampl = tools.ampl_notebook(
    modules=["highs"], # modules to install
    license_uuid="default") # license to use
```

```
Using default Community Edition License for Colab. Get yours at: https://ampl.com/ce
Licensed to AMPL Community Edition License for the AMPL Model Colaboratory (https://colab.ampl.com).
```

# Example: Network design with redundancy ([https://colab.ampl.com](https://colab.ampl.com))

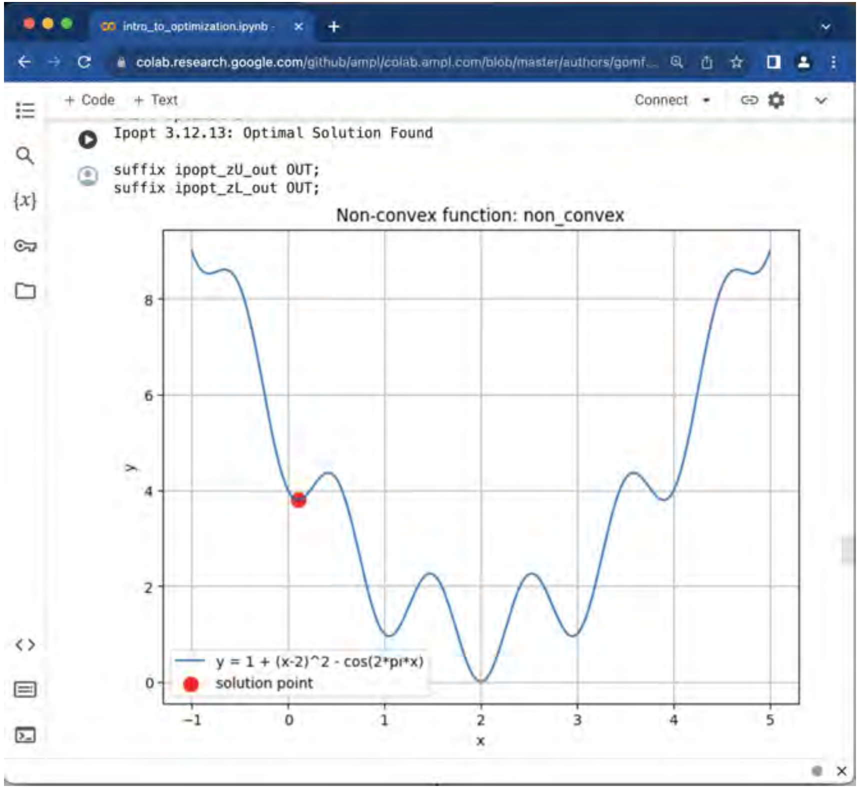

Notebook stored in GitHub
ic_grid_with_redundan
colab.research.google.com/github/ampl/colab.ampl.com/blob/master/authors/fdabrandao/...
electric_grid_with_redundancy.ipynb
File Edit View Insert Runtime Tools Help
Share
+ Code + Text
Copy to Drive
Connect

```
param cost{L};        # cost to build a link
param capacity;       # capacity of links

var Build {L} binary;   # Build[i,j] = 1 iff link btw i & j is built
var Flow {A} >= 0;      # Flow[i,j] is flow from i to j

minimize TotalBuildCost:
    sum {(i,j) in L} cost[i,j] * Build[i,j];

subject to Balance {i in N}:
    sum {(j,i) in A} Flow[j,i] - sum{(i,j) in A} Flow[i,j] >= demsup[i];

subject to ArcExists1 {(i,j) in L}:
    Flow[i,j] <= Build[i,j] * capacity;
subject to ArcExists2 {(i,j) in L}:
    Flow[j,i] <= Build[i,j] * capacity;

var FlRm {A,N} >= 0; # FlRm[i,j,rn] is flow from i to j when node rn is removed

subject to RemoveNode {rm in N}:
    sum {(i,rm) in A} FlRm[i,rm,rm] + sum {(rm,j) in A} FlRm[rm,j,rm] = 0;

subject to BalanceRm {i in N, rm in N: i != rm}:
    sum {(j,i) in A} FlRm[j,i,rm] - sum {(i,j) in A} FlRm[i,j,rm] >= demsup[i];

subject to ArcExistsRm1 {(i,j) in L, rm in N}:
    FlRm[i,j,rm] <= Build[i,j] * capacity;
subject to ArcExistsRm2 {(i,j) in L, rm in N}:
    FlRm[j,i,rm] <= Build[i,j] * capacity;
```


electric_grid_with_redundan
colab.research.google.com/github/ampl/colab.ampl.com/blob/master/authors/fdabrandao/...
electric_grid_with_redundancy.ipynb
File Edit View Insert Runtime Tools Help
Share
+ Code + Text
Copy to Drive
Connect
Default state
650
550
400
150
250
0
250
0
0
250
450
0
300
300
0
500
500
750
950
50
350

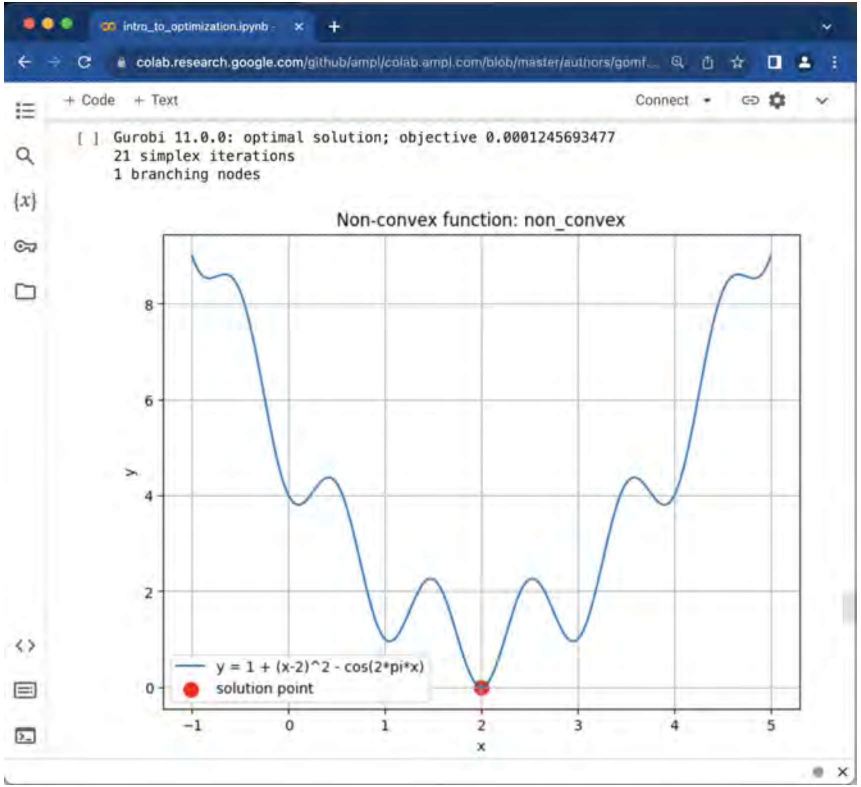
```
draw_solution(
    ampl.get_data("{(i,j) in A: Build[min(i,j),max(i,j)] > 0} Flow[i,j]").to_dict(),
    "Default state",
)
```

# Global Optimization with Gurobi ([https://colab.ampl.com](https://colab.ampl.com))



```
Ipopt 3.12.13: Optimal Solution Found

suffix ipopt_zU_out OUT;
suffix ipopt_zL_out OUT;
```

intro_to_optimization.ipynb
colab.research.google.com/github/ampl/colab.ampl.com/blob/master/authors/gomf...
+ Code
+ Text
Connect
Non-convex function: non_convex
y
x
0
2
4
6
8
−1
0
1
2
3
4
5
y = 1 + (x-2)^2 - cos(2*pi*x)
solution point



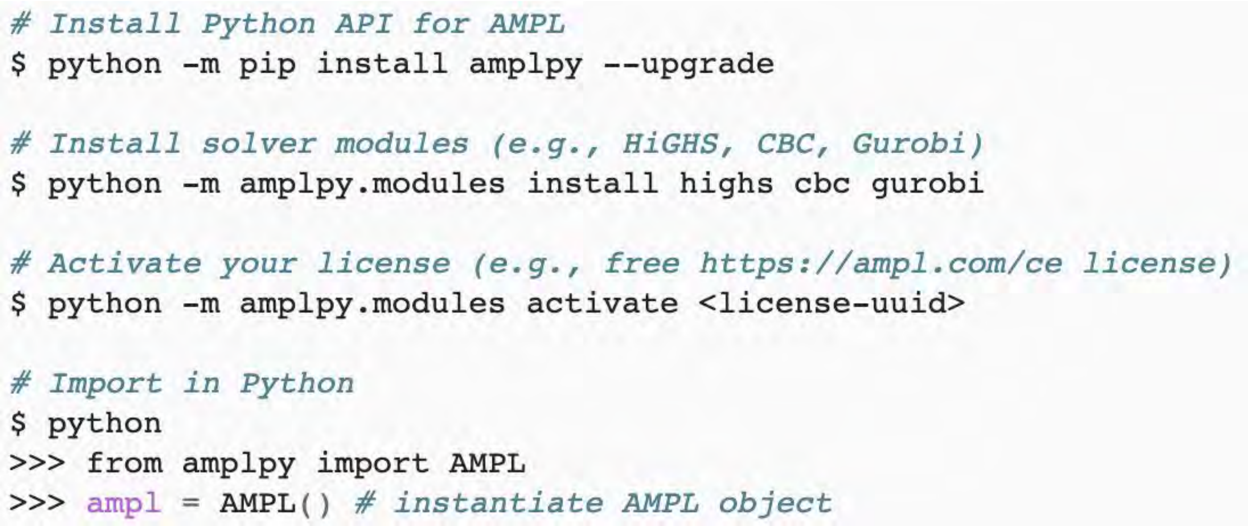
```
[ ] Gurobi 11.0.0: optimal solution; objective 0.0001245693477
    21 simplex iterations
    1 branching nodes
```

intro_to_optimization.ipynb
colab.research.google.com/github/ampl/colab.ampl.com/blob/master/authors/gomf...
+ Code
+ Text
Connect
Non-convex function: non_convex
y
x
0
2
4
6
8
−1
0
1
2
3
4
5
y = 1 + (x-2)^2 - cos(2*pi*x)
solution point

# Wait a minute. How are AMPL & solvers running on Google Colab integrated with Python 🐍?

# AMPL and all Solvers are now available as Python Packages

AMPL and all solvers are now available as python packages for **Windows**, **Linux** (**X86_64**, **aarch64**, **ppc64le**), and **macOS (Intel, Apple Silicon)**.



```
# Install Python API for AMPL
$ python -m pip install amplpy --upgrade

# Install solver modules (e.g., HiGHS, CBC, Gurobi)
$ python -m amplpy.modules install highs cbc gurobi

# Activate your license (e.g., free https://ampl.com/ce license)
$ python -m amplpy.modules activate <license-uuid>

# Import in Python
$ python
>>> from amplpy import AMPL
>>> ampl = AMPL() # instantiate AMPL object
```

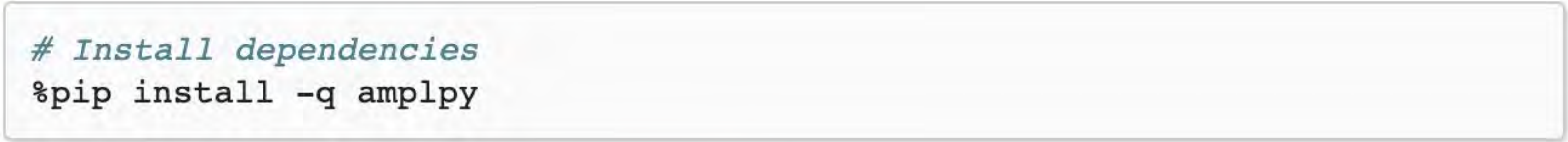
> https://ampl.com/python/

# AMPL is Free on Google Colab

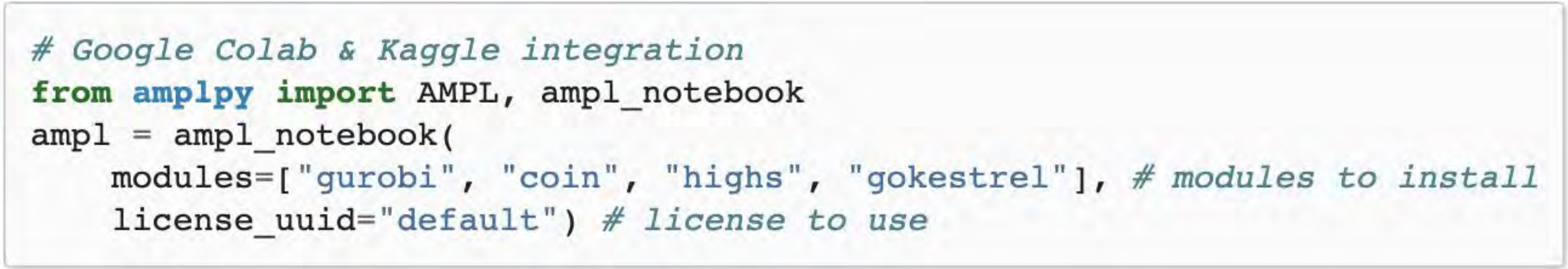
> [https://dev.ampl.com/ampl/python/colab.html](https://dev.ampl.com/ampl/python/colab.html)

> [https://try.ampl.com](https://try.ampl.com) (quickly access to AMPL on Colab)

You can install AMPL on Google Colab (where it is free by default) as follows:



```
# Install dependencies
%pip install -q amplpy
```



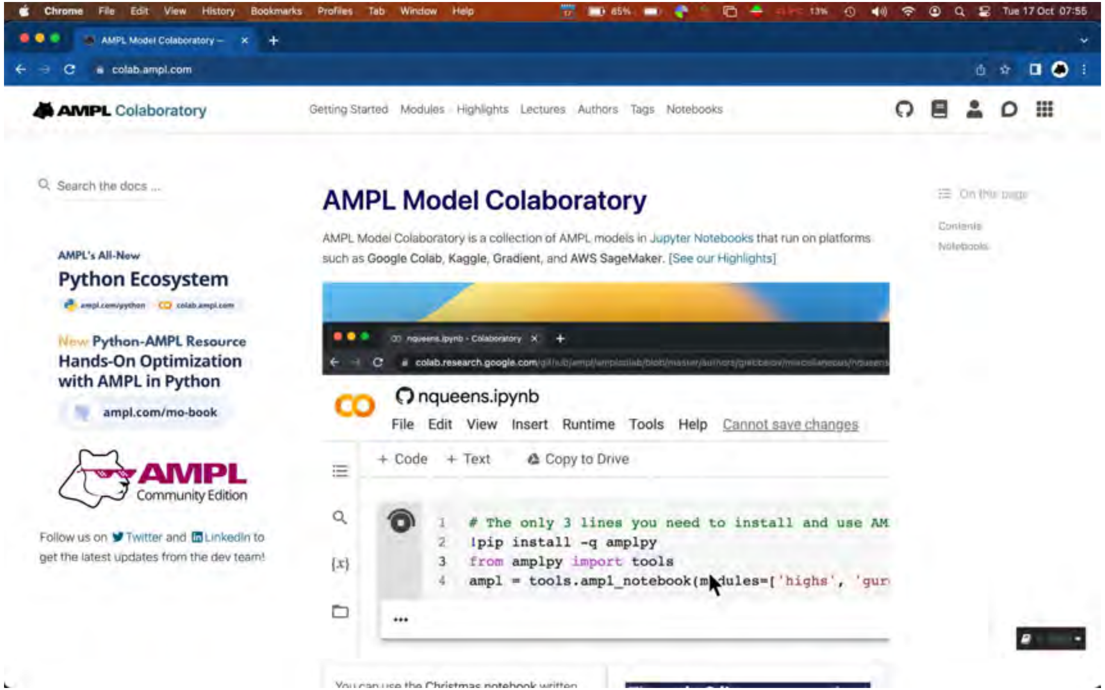
```
# Google Colab & Kaggle integration
from amplpy import AMPL, ampl_notebook
ampl = ampl_notebook(
    modules=["gurobi", "coin", "highs", "gokestrel"], # modules to install
    license_uuid="default") # license to use
```

# The Python-first 🐍 approach to learn and model with AMPL!

# AMPL Model Colaboratory ([https://colab.ampl.com](https://colab.ampl.com))

---

> Many examples: [https://colab.ampl.com](https://colab.ampl.com) (live demo)



## AMPL Model Colaboratory

AMPL Model Colaboratory is a collection of AMPL models in Jupyter Notebooks that run on platforms such as Google Colab, Kaggle, Gradient, and AWS SageMaker. [See our Highlights]

nqueens.ipynb

```
# The only 3 lines you need to install and use AM
!pip install -q amplpy
from amplpy import tools
ampl = tools.ampl_notebook(modules=['highs', 'gur
```

...

You can use the Christmas notebook written

AMPL's All-New

**Python Ecosystem**

ampl.com/python colab.ampl.com

New Python-AMPL Resource

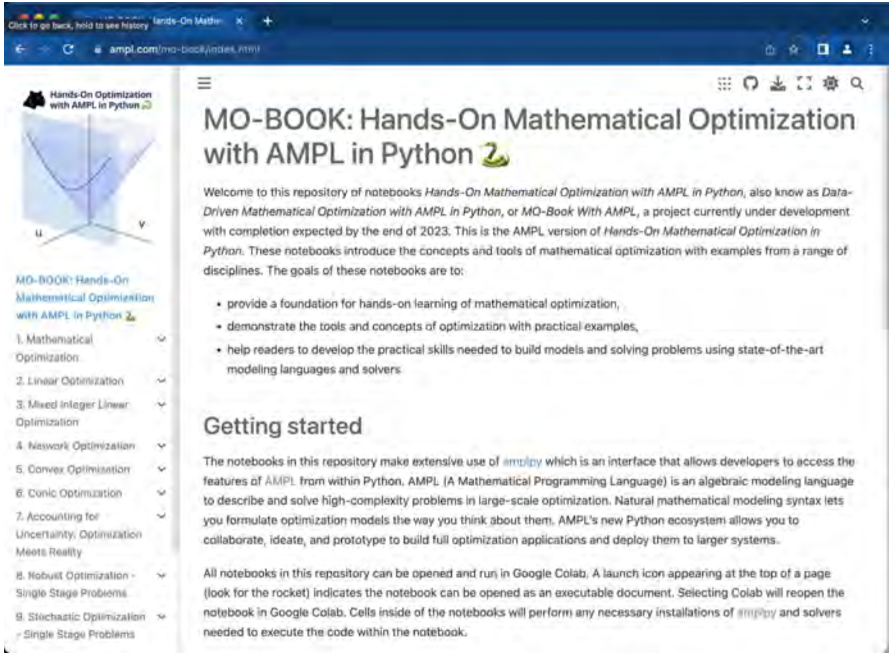
**Hands-On Optimization with AMPL in Python**

ampl.com/mo-book

Follow us on Twitter and LinkedIn to get the latest updates from the dev team!

# Hands-On Mathematical Optimization with AMPL in Python 🐍

> New Book: https://ampl.com/mo-book (live demo)


Click to go back, hold to see history
Hands-On Mathe
ampl.com/mo-book/index.html
Hands-On Optimization with AMPL in Python
u
v

MO-BOOK: Hands-On Mathematical Optimization with AMPL in Python 🐍
1. Mathematical Optimization
2. Linear Optimization
3. Mixed Integer Linear Optimization
4. Network Optimization
5. Convex Optimization
6. Conic Optimization
7. Accounting for Uncertainty: Optimization Meets Reality
8. Robust Optimization - Single Stage Problems
9. Stochastic Optimization - Single Stage Problems

### MO-BOOK: Hands-On Mathematical Optimization with AMPL in Python 🐍

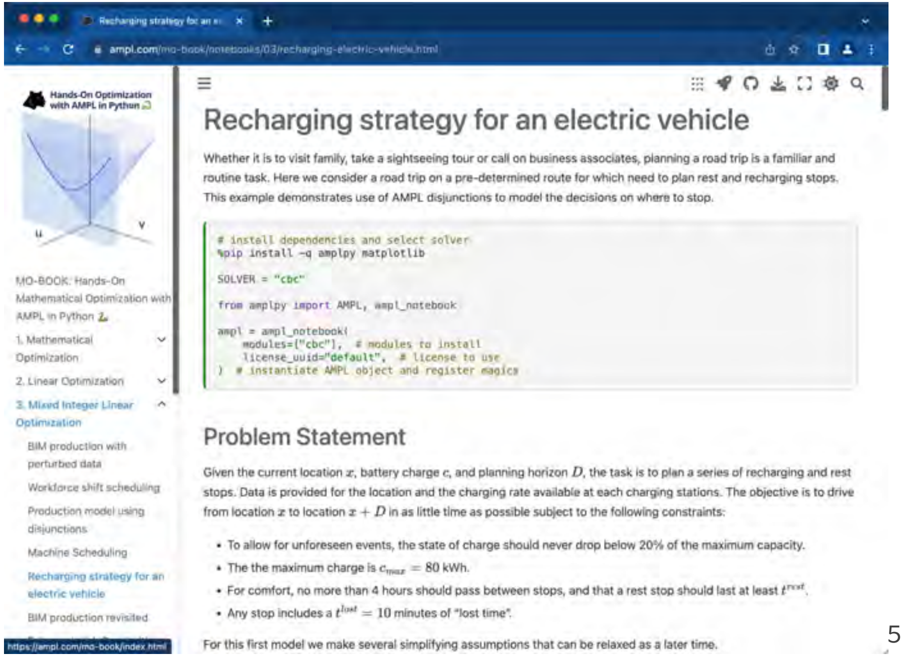
Welcome to this repository of notebooks *Hands-On Mathematical Optimization with AMPL in Python*, also know as *Data-Driven Mathematical Optimization with AMPL in Python*, or *MO-Book With AMPL*, a project currently under development with completion expected by the end of 2023. This is the AMPL version of *Hands-On Mathematical Optimization in Python*. These notebooks introduce the concepts and tools of mathematical optimization with examples from a range of disciplines. The goals of these notebooks are to:

- provide a foundation for hands-on learning of mathematical optimization,
- demonstrate the tools and concepts of optimization with practical examples,
- help readers to develop the practical skills needed to build models and solving problems using state-of-the-art modeling languages and solvers.

### Getting started

The notebooks in this repository make extensive use of amplpy which is an interface that allows developers to access the features of AMPL from within Python. AMPL (A Mathematical Programming Language) is an algebraic modeling language to describe and solve high-complexity problems in large-scale optimization. Natural mathematical modeling syntax lets you formulate optimization models the way you think about them. AMPL's new Python ecosystem allows you to collaborate, ideate, and prototype to build full optimization applications and deploy them to larger systems.

All notebooks in this repository can be opened and run in Google Colab. A launch icon appearing at the top of a page (look for the rocket) indicates the notebook can be opened as an executable document. Selecting Colab will reopen the notebook in Google Colab. Cells inside of the notebooks will perform any necessary installations of amplpy and solvers needed to execute the code within the notebook.


Recharging strategy for an e
ampl.com/mo-book/notebooks/03/recharging-electric-vehicle.html
Hands-On Optimization with AMPL in Python
u
v
https://ampl.com/mo-book/index.html

MO-BOOK: Hands-On Mathematical Optimization with AMPL in Python 🐍
1. Mathematical Optimization
2. Linear Optimization
3. Mixed Integer Linear Optimization
  BIM production with perturbed data
  Workforce shift scheduling
  Production model using disjunctions
  Machine Scheduling
  Recharging strategy for an electric vehicle
  BIM production revisited

### Recharging strategy for an electric vehicle

Whether it is to visit family, take a sightseeing tour or call on business associates, planning a road trip is a familiar and routine task. Here we consider a road trip on a pre-determined route for which need to plan rest and recharging stops. This example demonstrates use of AMPL disjunctions to model the decisions on where to stop.

```
# install dependencies and select solver
%pip install -q amplpy matplotlib

SOLVER = "cbc"

from amplpy import AMPL, ampl_notebook

ampl = ampl_notebook(
    modules=["cbc"],  # modules to install
    license_uuid="default",  # license to use
)  # instantiate AMPL object and register magics
```

### Problem Statement

Given the current location $x$, battery charge $c$, and planning horizon $D$, the task is to plan a series of recharging and rest stops. Data is provided for the location and the charging rate available at each charging stations. The objective is to drive from location $x$ to location $x + D$ in as little time as possible subject to the following constraints:

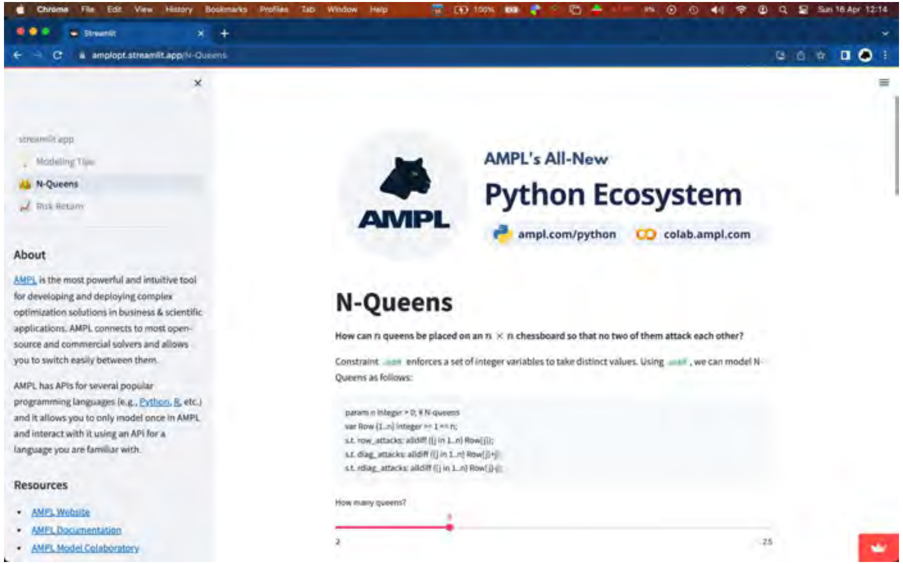
- To allow for unforeseen events, the state of charge should never drop below 20% of the maximum capacity.
- The the maximum charge is $c_{max} = 80$ kWh.
- For comfort, no more than 4 hours should pass between stops, and that a rest stop should last at least $t^{rest}$.
- Any stop includes a $t^{lost} = 10$ minutes of "lost time".

For this first model we make several simplifying assumptions that can be relaxed as a later time.

# Deploying optimization applications quickly and easily using AMPL with Python 🐍

# AMPL on Streamlit

- [https://ampl.com/streamlit](https://ampl.com/streamlit) (live demo)



**About**

AMPL is the most powerful and intuitive tool for developing and deploying complex optimization solutions in business & scientific applications. AMPL connects to most open-source and commercial solvers and allows you to switch easily between them.

AMPL has APIs for several popular programming languages (e.g., Python, R, etc.) and it allows you to only model once in AMPL and interact with it using an API for a language you are familiar with.

**Resources**

- AMPL Website
- AMPL Documentation
- AMPL Model Colaboratory

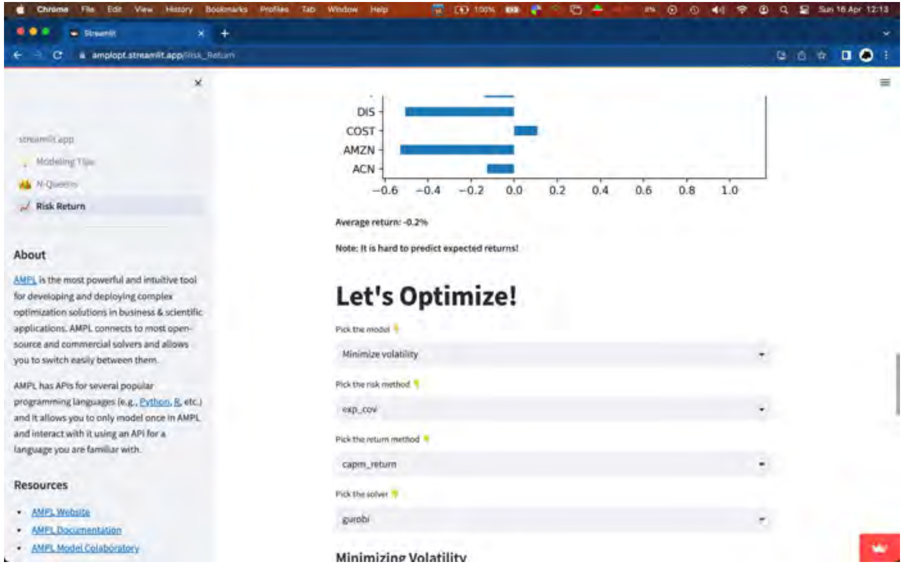
## N-Queens

**How can $n$ queens be placed on an $n \times n$ chessboard so that no two of them attack each other?**

Constraint [illegible] enforces a set of integer variables to take distinct values. Using [illegible], we can model N-Queens as follows:

```
param n integer > 0; # N-queens
var Row {1..n} integer >= 1 <= n;
s.t. row_attacks: alldiff ({j in 1..n} Row[j]);
s.t. diag_attacks: alldiff ({j in 1..n} Row[j]+
s.t. rdiag_attacks: alldiff ({j in 1..n} Row[j]-
```

How many queens?



**About**

AMPL is the most powerful and intuitive tool for developing and deploying complex optimization solutions in business & scientific applications. AMPL connects to most open-source and commercial solvers and allows you to switch easily between them.

AMPL has APIs for several popular programming languages (e.g., Python, R, etc.) and it allows you to only model once in AMPL and interact with it using an API for a language you are familiar with.

**Resources**

- AMPL Website
- AMPL Documentation
- AMPL Model Colaboratory

**Average return: -0.2%**

**Note: It is hard to predict expected returns!**

## Let's Optimize!

Pick the model: Minimize volatility

Pick the risk method: exp_cov

Pick the return method: capm_return

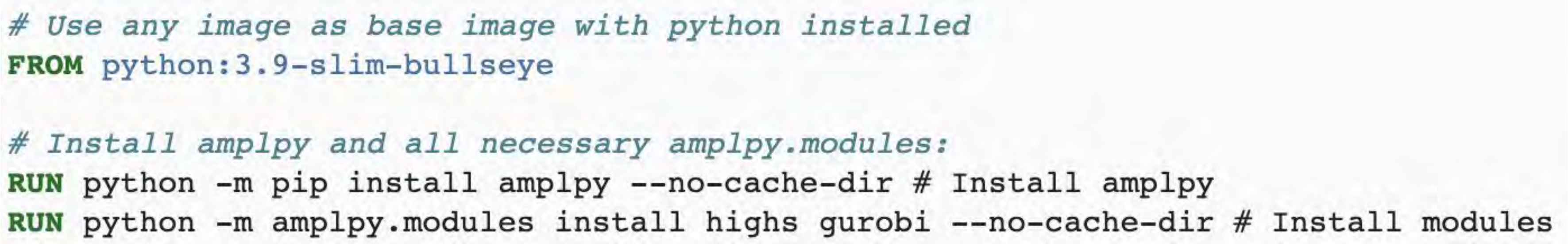
Pick the solver: gurobi

### Minimizing Volatility

# Deploy anywhere with Docker

› https://dev.ampl.com/ampl/docker/

AMPL can be easily used on Docker containers and deployed anywhere.



```
# Use any image as base image with python installed
FROM python:3.9-slim-bullseye

# Install amplpy and all necessary amplpy.modules:
RUN python -m pip install amplpy --no-cache-dir # Install amplpy
RUN python -m amplpy.modules install highs gurobi --no-cache-dir # Install modules
```

# Example project showing how to deploy applications

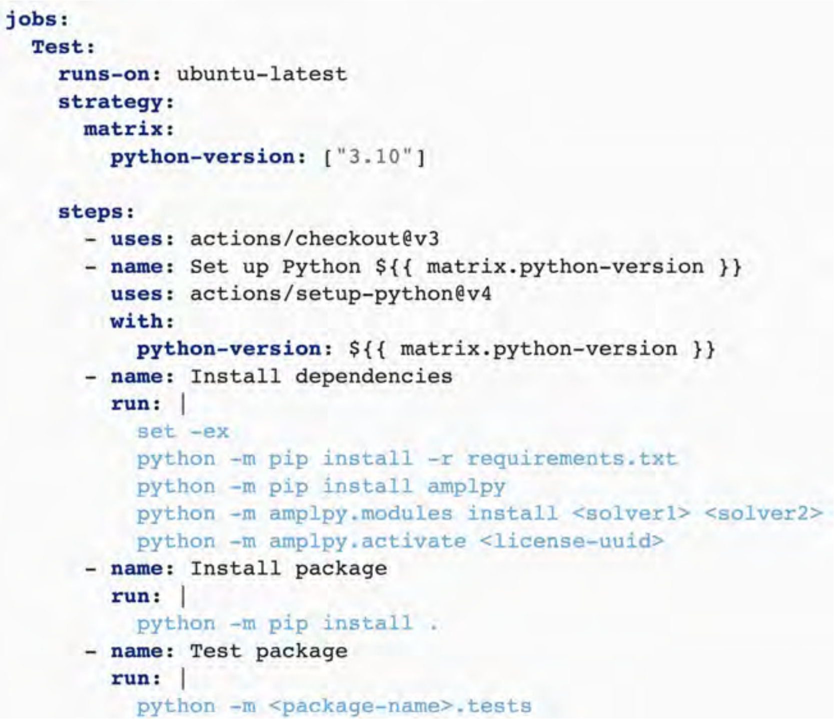
- https://amplpyfinance.ampl.com/
  - How to use AMPL with Docker Containers:
    - A basic Docker Compose template for orchestrating a **Flask** application & a **Celery** queue with **Redis**.
    - https://github.com/ampl/amplpyfinance/tree/master/deployment/docker

  - The same Docker images can be deployed to **Kubernetes Clusters**

  - How to use AMPL in Continuous Integration Systems
    - This project uses **Azure Pipelines** and **GitHub Actions** for **CI/CD**
    - https://dev.ampl.com/ampl/cicd/

# Continuous Integration Systems

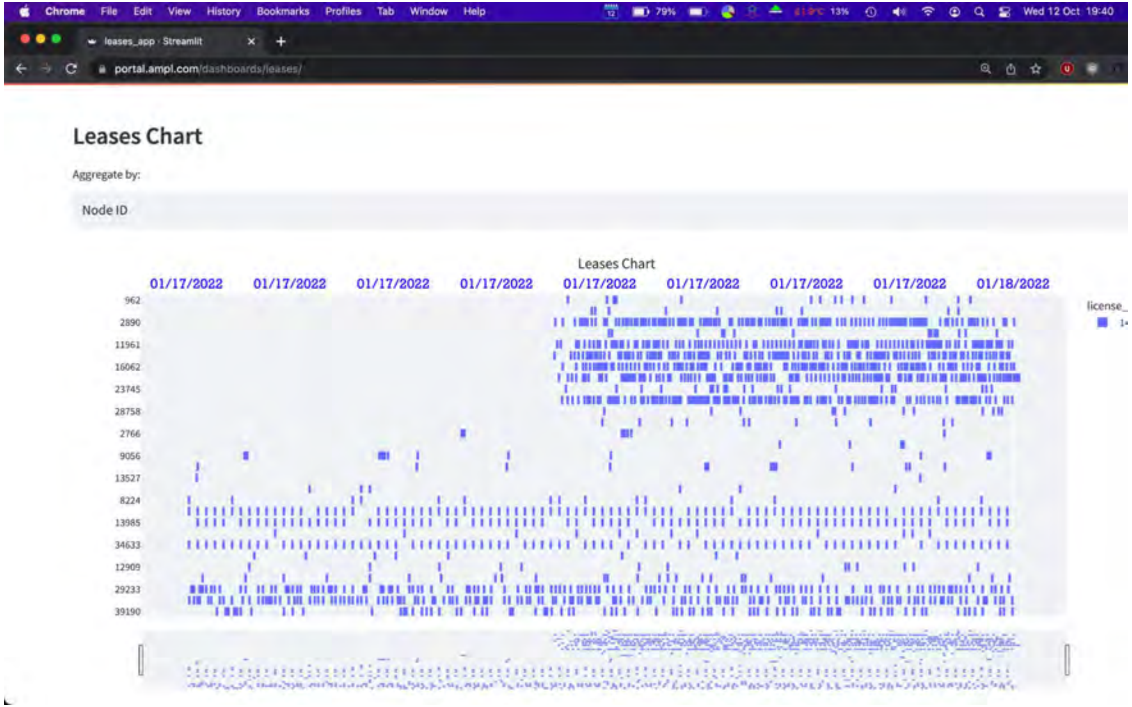
- How to use AMPL in Continuous Integration Systems
  - This project uses **Azure Pipelines** and **GitHub Actions** for **CI/CD**
  - [https://dev.ampl.com/ampl/cicd/](https://dev.ampl.com/ampl/cicd/)



```
jobs:
  Test:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: ["3.10"]

    steps:
      - uses: actions/checkout@v3
      - name: Set up Python ${{ matrix.python-version }}
        uses: actions/setup-python@v4
        with:
          python-version: ${{ matrix.python-version }}
      - name: Install dependencies
        run: |
          set -ex
          python -m pip install -r requirements.txt
          python -m pip install amplpy
          python -m amplpy.modules install <solver1> <solver2>
          python -m amplpy.activate <license-uuid>
      - name: Install package
        run: |
          python -m pip install .
      - name: Test package
        run: |
          python -m <package-name>.tests
```

# What about licenses for AMPL and Commercial Solvers?

# Dynamic Licensing System



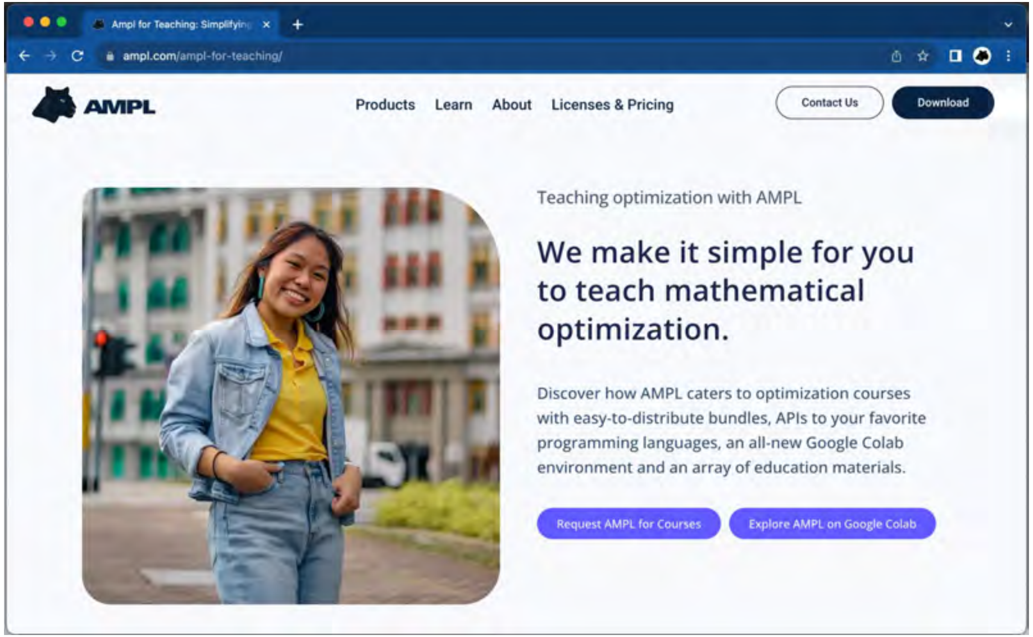
## Leases Chart

Aggregate by:

Node ID

# Free licenses to use on Google Colab (and locally!)

- **ampl.com/ce**
  - For personal use
  - **Immediate access without approvals required!**
  - No size-limits
  - Includes access to:
    - Open-source solvers
    - Commercial solver trials
- **ampl.com/courses**
  - For teaching
  - No size-limits
  - **Full access to all solvers!**
  - All students can use the license during the course.


Ampl for Teaching: Simplifying
ampl.com/ampl-for-teaching/
AMPL
Products
Learn
About
Licenses & Pricing
Contact Us
Download
Request AMPL for Courses
Explore AMPL on Google Colab

Teaching optimization with AMPL

We make it simple for you to teach mathematical optimization.

Discover how AMPL caters to optimization courses with easy-to-distribute bundles, APIs to your favorite programming languages, an all-new Google Colab environment and an array of education materials.

# Learn more

---

- **https://ampl.com/mo-book**
  - New AMPL+Python Book!
- **https://ampl.com/streamlit**
  - Streamlit App with many examples
- **https://colab.ampl.com**
  - Collection of AMPL models in Jupyter Notebooks
- **https://amplpy.ampl.com**
  - Python API Documentation
- **https://mp.ampl.com/model-guide.html**
  - Modeling Guide for MP-based AMPL Solvers
- **https://ampl.com/courses**
  - Free license for teaching